UNITED STATES PATENT APPLICATION

for

~~METHOD AND APPARATUS FOR MAINTAINING THE TEMPERATURE OF A MICROPROCESSOR USING A FIRMWARE ASSISTED INTERRUPT MECHANISM~~

Inventors:

Ganesh Subramaniyam
Gary Hammond

prepared by:

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP
12400 Wilshire Boulevard
Los Angeles, CA 90025-1026
(408) 720-8300

File No.: 042390.P9043

"Express Mail" mailing label number EL617210509US
Date of Deposit September 25, 2000
I hereby certify that this paper or fee is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to the Commissioner of Patents and Trademarks, Washington, D.C. 20231.

Lori Cox
(Typed or printed name of person mailing paper or fee)

Lori Cox
(Signature of person mailing paper or fee)

METHOD AND APPARATUS FOR MAINTAINING THE TEMPERATURE OF A MICROPROCESSOR USING A FIRMWARE ASSISTED INTERRUPT MECHANISM

FIELD OF THE INVENTION

The present invention relates to computer systems; more particularly, the present invention relates to lowering and maintaining the temperature of a microprocessor die below a burnout temperature.

BACKGROUND

Throughout the history of microcomputers there has been a motivation to increase the performance of microprocessors. However, with the constant increase in microprocessor performance, there is typically an increase in the magnitude of power consumed by the microprocessor. Due to the increase in power consumption, the run time temperature of the die of a microprocessor may exceed a safe threshold value.

Various methods currently exist to reduce the run time temperature of microprocessors. One such method is to modulate the processor clock. Another method is to modulate the processor clock frequency. However, these methods complicate the hardware design implementation, validation and also decrease the performance of a microprocessor. Yet another solution for cooling the run time temperature of a microprocessor is to shut down the microprocessor and reboot the computer system at a later time. However having to shut down the computer is obviously disadvantageous as it increases the down time of the system. Therefore, it would be advantageous to develop a more efficient method of maintaining the run time temperature of a microprocessor.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention will be understood more fully from the detailed description given below and from the accompanying drawings of various embodiments of the invention. The drawings, however, should not be taken to limit the invention to the specific embodiments, but are for explanation and understanding only.

Figure 1 is a block diagram of one embodiment of a computer system;

Figure 2 is a block diagram of one embodiment of a microprocessor; and

Figure 3 is a flow diagram for one embodiment of controlling the temperature of a microprocessor.

DETAILED DESCRIPTION

A method and apparatus for maintaining the temperature of a microprocessor is described. In the following detailed description of the present invention numerous specific details are set forth in order to provide a thorough understanding of the present invention. However, it will be apparent to one skilled in the art that the present invention may be practiced without these specific details. In other instances, well-known structures and devices are shown in block diagram form, rather than in detail, in order to avoid obscuring the present invention.

Reference in the specification to "one embodiment" or "an embodiment" means that a particular feature, structure, or characteristic described in connection with the embodiment is included in at least one embodiment of the invention. The appearances of the phrase "in one embodiment" in various places in the specification are not necessarily all referring to the same embodiment.

Figure 1 is a block diagram of one embodiment of a computer system 100. Computer system 100 includes a central processing unit (processor) 105 coupled to processor bus 110. In one embodiment, processor 105 is a processor in the Pentium® family of processors including the Pentium® II family and mobile Pentium® and Pentium® II processors available from Intel Corporation of Santa Clara, California. Alternatively, other processors may be used. Processor 105 may include a first level (L1) cache memory (not shown in **Figure 1**).

According to one embodiment, processor 105 operates in either a full dispersal mode or a single dispersal mode. In the full dispersal mode, processor 105 executes multiple instructions at a time. According to one embodiment, processor 105 executes six instructions at a time. In the single dispersion mode, processor 105 executes one instruction at a time. According to a further

embodiment, processor 105 transitions from the full dispersion mode to the single dispersion mode upon the die temperature of processor 105 exceeding a predetermined temperature threshold.

In yet a further embodiment, processor 105 operates according to an artificial activity mode. The artificial activity mode minimizes current spikes (e.g., $\frac{di}{dt}$ spikes) within processor 105 by maintaining a minimum level of activity within processor 105. For example, if the activity (e.g., instructions received and/or executed) falls below a predetermined threshold, simulated instructions are received at processor 105 for processing. The simulated instructions may be received from the L1 cache memory, a floating point unit, integer unit or any other device within processor 105 or computer system 100. The results of the simulated instructions are disregarded after processing. According to one embodiment, the minimum level of activity within processor 105 is seventy percent of processor 105 capacity. However in other embodiments, the minimum level of activity within processor 105 may be other percentages of processor 105 capacity.

In one embodiment, processor 105 is also coupled to cache memory 107, which is a second level (L2) cache memory, via dedicated cache bus 102. The L1 and L2 cache memories can also be integrated into a single device. Alternatively, cache memory 107 may be coupled to processor 105 by a shared bus. Cache memory 107 is optional and is not required for computer system 100.

Chip set 120 is also coupled to processor bus 110. In one embodiment, chip set 120 is the 440BX chip set available from Intel Corporation; however, other chip sets can also be used. Chip set 120 may include a memory controller for controlling a main memory 113. Further, chipset 220 may also include an Accelerated Graphics Port (AGP) Specification Revision 2.0 interface 320

developed by Intel Corporation of Santa Clara, California. AGP interface 320 is coupled to a video device 125 and handles video data requests to access main memory 113.

Main memory 113 is coupled to processor bus 110 through chip set 120. Main memory 113 and cache memory 107 store sequences of instructions that are executed by processor 105. The sequences of instructions executed by processor 105 may be retrieved from main memory 113, cache memory 107, or any other storage device. Additional devices may also be coupled to processor bus 110, such as multiple processors and/or multiple main memory devices. Computer system 100 is described in terms of a single processor; however, multiple processors can be coupled to processor bus 110. Video device 125 is also coupled to chip set 120. In one embodiment, video device 125 includes a video monitor such as a cathode ray tube (CRT) or liquid crystal display (LCD) and necessary support circuitry.

Processor bus 110 is coupled to system bus 130 by chip set 120. In one embodiment, system bus 130 is a Peripheral Component Interconnect (PCI) bus adhering to a Specification Revision 2.1 bus developed by the PCI Special Interest Group of Portland, Oregon; however, other bus standards may also be used. Multiple devices, such as audio device 127, may be coupled to system bus 130.

Bus bridge 140 couples system bus 130 to secondary bus 150. In one embodiment, secondary bus 150 is an Industry Standard Architecture (ISA) Specification Revision 1.0a bus developed by International Business Machines of Armonk, New York. However, other bus standards may also be used, for example Extended Industry Standard Architecture (EISA) Specification Revision 3.12 developed by Compaq Computer, et al. Multiple devices, such as hard disk 153 and disk drive 154 may be coupled to secondary bus 150. Other devices, such as cursor control devices (not shown in **Figure 1**), may be coupled to

secondary bus 150.

According to one embodiment, a basic input output system (BIOS) 155 is coupled to secondary bus 150. BIOS 155 includes arrays of programmable AND gates and predefined OR gates that store a set of routines which provide an interface between the operating system and components of computer system 100. According to one embodiment, BIOS 155 transmits signals to processor 105 to initiate the generation of artificial activity at processor 105. In one embodiment, BIOS 155 is programmable array logic (PAL). However, one of ordinary skill in the art will appreciate that other devices may be used to implement BIOS 155.

According to one embodiment, processor 105 includes power management logic to prevent prolonged operation at excess temperatures. During the run time of computer system 105, the power consumed at processor 105 may exceed 130 watts. Such power consumption may cause processor 105 to overheat and lead to the eventual burnout of processor 105. **Figure 2** is a block diagram of one embodiment of temperature monitoring logic within processor 105.

Referring to **Figure 2**, processor 105 includes a thermal sensor 210, an analog to digital converter (ADC) 220, a filter 230, interrupt generating hardware 240, instruction execution unit 250, and artificial activity generator 260. In addition, processor 105 is coupled to an interrupt handler 270. According to one embodiment, sensor 210 is an analog sensor that continuously monitors the temperature of processor 105 during the operation of computer system 100. ADC 220 is coupled to sensor 210 and converts an analog temperature value received from sensor 210 to a one-bit digital signal.

According to one embodiment, ADC 220 transmits a low logic level (e.g., logic 0) if the temperature value received is below a predetermined threshold and transmits a high logic level (e.g., logic 1) if the temperature value is above

the predetermined threshold. One of ordinary skill in the art will appreciate that the combination of sensor 210 and ADC 220 may be replaced by a digital sensor in other embodiments.

Filter 230 is coupled to ADC 220. Filter 230 is a digital filter that removes temperature noise conditions for a predetermined number of clock cycles. According to one embodiment, filter 230 determines how long the die temperature is above or below the predetermined threshold before initiating a high temperature or normal temperature interrupt, respectively. According to a further embodiment, digital filter 230 removes noise conditions for two clock cycles. In yet a further embodiment, the number of predetermined clock cycles may be programmed into digital filter 230.

Interrupt generating hardware 240 is coupled to filter 230. Interrupt generating hardware 240 generates a high temperature (HITEMP) interrupt upon the die temperature of processor 105 exceeding the predetermined threshold temperature, subject to the operations of filter 230. In addition, interrupt generating hardware 240 generates a normal temperature (NORMTEMP) interrupt upon the die temperature of processor 105 cooling below the predetermined threshold temperature. In one embodiment, a high logic level is transmitted by interrupt generating hardware 240 as the HITEMP interrupt. Further, a low logic level is transmitted by interrupt generating hardware 240 as the NORMTEMP interrupt. One of ordinary skill in the art will recognize that the operation of ADC 220 may be reversed.

Instruction execution unit 250 determines the dispersal mode in which processor 105 operates. In one embodiment, instruction execution unit 250 causes processor 105 to operate in the full dispersal mode whenever the die temperature is below the predetermined threshold temperature. Conversely, execution unit 250 causes processor 105 to operate in the single dispersal mode

whenever the die temperature is above the predetermined threshold temperature.

Artificial activity generator 260 controls the artificial activity within processor 105. As described above, an artificial activity mode minimizes current spikes within processor 105 by maintaining a minimum level of activity. Artificial activity generator 260 determines the level of artificial activity that is generated at processor 105. According to one embodiment, artificial activity generator suspends artificial activity within processor 105 whenever the die temperature is above the predetermined threshold temperature.

Interrupt handler 270 is coupled to interrupt generating hardware 240, instruction execution unit 250 and artificial activity generator 260. In one embodiment, interrupt handler 270 receives the processor level interrupts HITEMP and NORMTEMP and causes the appropriate action to be taken. For example, upon receiving the HITEMP interrupt, interrupt handler 270 transmits a signal to instruction execution unit 250 causing processor 105 to transition from the full dispersal mode to the single dispersal mode. By placing processor 105 in the single dispersal mode, the utilization of components within processor 105 is reduced, resulting in the cooling of temperature within processor 105. Similarly, interrupt handler 270 causes processor 105 to transition back to the full dispersal mode upon receiving the NORMTEMP interrupt.

Further, interrupt handler 270 transmits signals to artificial activity generator to suspend and resume artificial activity depending upon the die temperature. According to one embodiment, interrupt handler 270 resides within BIOS 155. In a further embodiment, interrupt handler 270 may resides in main memory 113 upon startup of computer system 100. However, one of ordinary skill in the art will appreciate that interrupt handler 270 may be located elsewhere within computer system 100.

Figure 3 is a flow diagram for one embodiment of controlling the temperature of processor 105. At process block 305, processor 105 is operating in the full dispersal mode. As described above, the full dispersal mode features executing instruction streams at very high processor 105 utilization. At process block 310, it is determined whether the die temperature of processor 105 has exceeded the predetermined threshold. If the die temperature has not exceeded the predetermined threshold, control is returned to process block 305.

However, if the die temperature has exceeded the predetermined threshold, the HITEMP interrupt is generated at ADC 220, process block 315. According to one embodiment, the execution of code within processor 105 is temporarily suspended after the HITEMP interrupt is generated. At process block 320, interrupt handler 270 causes processor 105 to cease operation in the artificial activity mode. By stopping artificial activity, processor 105 is permitted to fall below the predetermined minimum level of activity.

In addition, at process block 325, interrupt handler 270 causes processor 105 to transition from the full dispersal mode to the single dispersal mode. At process block 330, the execution of code within processor 105 continues in the single dispersion mode from the point at which it was suspended. As described above, the single dispersal mode clamps the maximum utilization of components within processor 105. As a result, the power consumed by processor 105 is limited. At process block 335, it is determined whether the die temperature of processor 105 continues to remain above the predetermined temperature threshold. If the temperature remains above the predetermined threshold, processor 105 is shut down, process block 340.

If, however, the die temperature of processor 105 falls below the predetermined threshold, the NORMTEMP interrupt is generated, process block 345. The execution of code within processor 105 is temporarily suspended after

the NORMTEMP interrupt is generated. At process block 350, interrupt service handler code within interrupt handler service 270 causes processor 105 to commence operation in the artificial activity mode. In addition, at process block 355, interrupt handler 270 causes processor 105 to transition from the single dispersal mode to the full dispersal mode. At process block 360, interrupt handler 270 causes the execution of code within processor 105 to continue in the full dispersion mode from the point at which it was suspended.

Whereas many alterations and modifications of the present invention will no doubt become apparent to a person of ordinary skill in the art after having read the foregoing description, it is to be understood that any particular embodiment shown and described by way of illustration is in no way intended to be considered limiting. Therefore, references to details of various embodiments are not intended to limit the scope of the claims which in themselves recite only those features regarded as the invention.